1. (Currently Amended) A system for loading software applications, comprising:

a server, executing a Java virtual machine that includes a system classloader, for storing and running a plurality of software applications, wherein each of said plurality of software applications includes a plurality of deployable modules and classes associated therewith, and wherein the software applications or modules therein can be customized by a software developer and then deployed to run on the same server;

a control file, that can be edited by the software developer and associated with said plurality of software applications or modules, wherein said control file specifies a hierarchy of application classloaders as children of the system classloader, to be used with the modules in said plurality of software applications, and wherein the hierarchy includes a plurality of nested branches and siblings, including providing each of said plurality of software applications or modules with its own application classloader hierarchy so that the software applications or modules are not aware of classloaders or classes that are assigned to another software application, and wherein the hierarchy is specified by the software developer to provide namespace separation between two or more of the plurality of software applications or between different modules in any one of the software applications, ***and wherein the system classloader includes only application level library classes, and wherein the nested branches of the hierarchy includes classes or modules that are frequently called by modules in said plurality of software applications as siblings within the same branch or same classloader***; and

a deployment utility that, upon receiving a request to deploy and run a software application on the server,

parses the control file and determines which classloaders are specified therein for the software application being deployed,

loads with said software application into the Java virtual machine at the server a selection of said application classloaders corresponding to the hierarchy specified by said control file, including creating sibling classloaders for implementations whose classes are to be loaded separately as specified by the control file, and, if a particular software application or a module in a software application is being redeployed then loading only the application classloaders that are specified in the branches for that particular software application or module, without loading any of the other branches in the hierarchy, and

enables the server to host multiple isolated software applications or modules within the Java virtual machine, as defined by the hierarchy;

wherein, upon receiving a call to instantiate an implementation, the system searches within the hierarchy including first searching the calling classloader, and then successively searching parent classloaders above the calling classloader, ~~***excluding any sibling classloaders,***~~ to locate and instantiate necessary classes or objects.